

HTML et JavaScript

Introduction

Nous l'avons vu précédemment, l'HTML organise le contenu d'une page, le CSS le met en forme, mais pour le moment, nous ne pouvons pas créer d'interaction entre les éléments.

Pour cela, nous allons utiliser le langage JavaScript.

Programmation événementielle

Avec Python, on a vu la programmation séquentielle. Le programme se déroule ligne par ligne, jusqu'à ce qu'il ait fini.

Pour les interfaces graphiques, le principe est différent. Nous allons faire de la programmation événementielle.

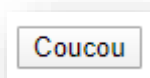
En HTML, chaque élément (un texte, une zone, un bouton...) va pouvoir réagir à un événement (un clic, un survol de la souris, une perte de focus, une utilisation du clavier...). Le principe de la programmation événementielle est de définir par la programmation comment va réagir l'élément face à cet événement.

TP 1 - Hello Word

1. Créez une page HTML « TP1.html » avec la structure minimum (`html`, `head`, `body`, `title`...)
2. Ajoutez un bouton dans la page avec le contenu suivant :

```
<button> Coucou </button>
```

3. Ouvrez le document dans un navigateur. Vous devez obtenir le résultat suivant



Si vous cliquez sur ce bouton il ne se passera rien. Nous allons lui dire comment réagir face à un événement.

4. Ajoutez au bouton l'attribut suivant :

```
onclick="alert('hello word !')"
```

Attention aux apostrophes et aux doubles guillemets. Le principe est le même en maths au collège avec les crochets et les parenthèses.

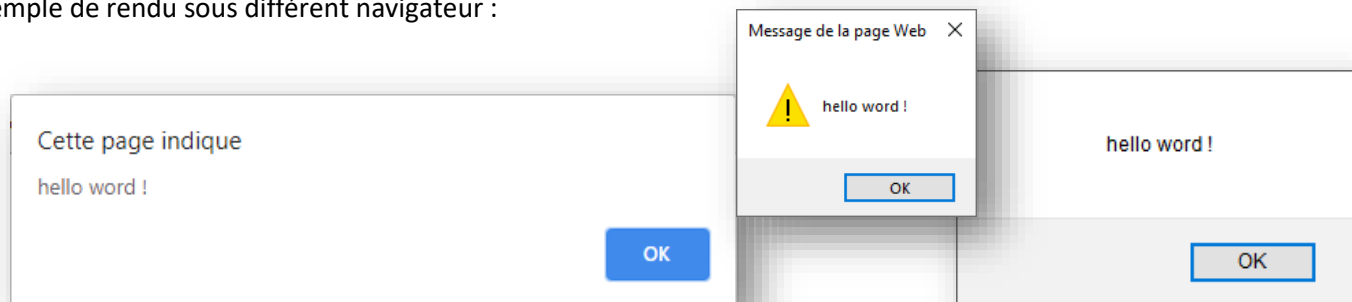
Vous devez donc avoir le code suivant :

```
<button onclick="alert('hello word !')"> Coucou </button>
```

5. Actualisez la page contenant votre bouton (raccourcis clavier : F5)
6. Cliquez sur le bouton.

Une fenêtre (différente en fonction des navigateurs) apparaît avec le texte « hello word ! »

Exemple de rendu sous différents navigateurs :



Bravo, vous avez fait votre première popup Javascript.

Différence entre JavaScript et Python

Il y a plusieurs différences importantes entre Javascript et Python.

- Les blocs de code ne sont pas définis grâce à l'indentation, mais en étant mis entre accolade. On peut ne pas respecter l'indentation, mais il est recommandé de le faire pour des raisons de lisibilité.
- Les lignes de code Javascript doivent (c'est du moins recommandé) se finir par un point virgule « ; »

Voici deux codes équivalents, un en Python, l'autre en Javascript :

```
# Python
def afficher(message):
    if (len(message)>3):
        print(message)
    else:
        print ("Message trop court")
```

```
// Javascript
function afficher(message){
    if (message.length >3){
        alert(message);
    }
    else{
        alert("Message trop court");
    }
}
```

Appel au code Javascript

Dans l'exemple précédent, nous avons exécuté précédent du code javascript. En effet, l'attribut `onclick` prend en paramètre directement du code javascript.

Mais si plusieurs boutons ont (presque) le même comportement, cela va être lourd à mettre en place. Nous allons donc, comme pour le CSS, centraliser le plus gros du code JavaScript.

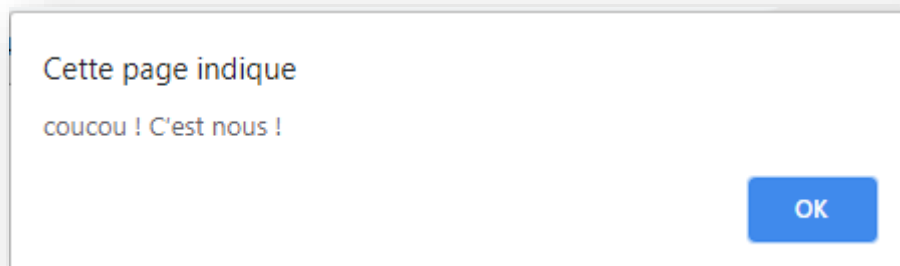
- 1) Dans la partie `head` de votre page HTML, ajoutez le bloc suivant

```
<script type='text/javascript'>
    function coucou() {
        alert("coucou ! C'est nous ! ");
    }
</script>
```

- 2) Modifiez l'attribut `onclick` du bouton précédent de la façon suivante :

```
onclick="coucou();">
```

- 3) Enregistrez, réactualisez la page internet et cliquez sur le bouton. Vous devez avoir le message suivant :



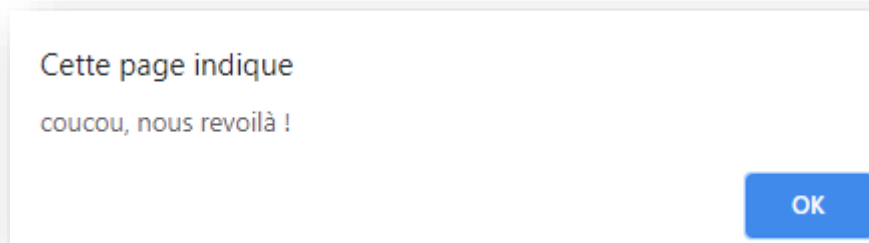
- 4) Modifiez la fonction javascript `coucou` pour avoir le code suivant :

```
<script>
    function coucou(mon_message) {
        alert(mon_message);
    }
</script>
```

5) Modifiez l'attribut `onclick` du bouton de la manière suivante :

```
onclick="coucou('coucou, nous revoilà !') ;"
```

6) Enregistrez, actualisez la page et cliquez sur le bouton



Gestion des erreurs JavaScript

1) Modifiez l'attribut `onclick` du bouton de la manière suivante :

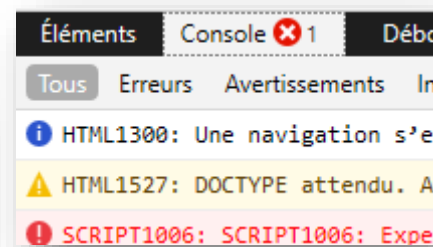
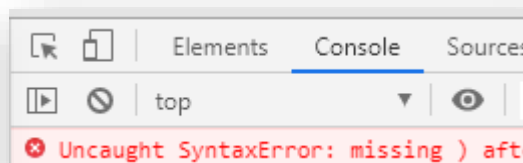
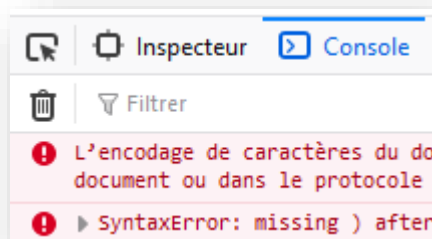
```
onclick="coucou('coucou, c'est encore nous !')"
```

2) Enregistrez, actualisez et cliquez.

Il ne se passe rien !

3) Utilisez le raccourci clavier « Ctrl + Shift + I » ou appuyez sur la touche « F12 » de votre clavier. Vous venez d'ouvrir les outils de développement.

4) Cliquez sur l'onglet « Console »



Vous pouvez remarquer la présence d'un message rouge indiquant la présence d'une erreur.

Certain l'auront peut être remarqué. L'erreur est dans le texte qu'on veut afficher. En effet, le texte est délimité comme en Python par des simples-guillemets (apostrophe) ou des doubles-guillemets.

Ici, on n'utilise les deux :

- un pour définir le code exécuté lors de l'événement onclick
- un autre pour définir le texte qui sera à afficher.

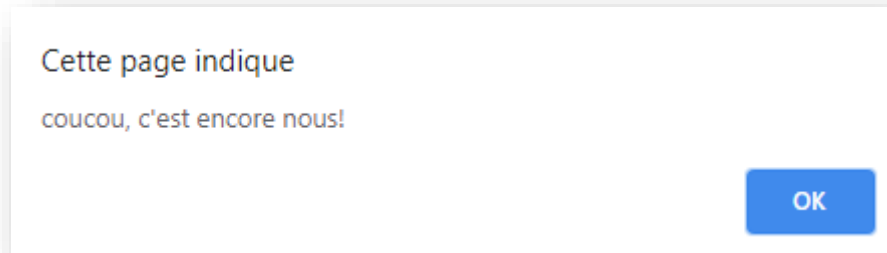
Or notre texte contient également une apostrophe. Javascript pense que le texte à afficher est « Coucou, c ». Par contre, il ne comprend pas ce que vient faire la suite « est encore nous ! » et ne sais pas comment le gérer. Il génère donc une erreur.

On va devoir utiliser un caractère d'échappement qui va permettre de considérer l'apostrophe comme une simple apostrophe, et non pas comme le caractère définition la fin de la chaîne de caractère. Comme dans beaucoup de langage, ce caractère est antislash « \ ».

5) Modifiez l'attribut `onclick` du bouton de la manière suivante :

```
onclick="coucou('coucou, c'est encore nous!')"
```

6) Enregistrez, réactualisez et cliquez



Résumé

- 1) Javascript est un langage de programmation pour ajouter des interactions qu'un site
- 2) On n'utilise pas des tabulations mais des accolades pour définir des blocs de code
- 3) Pour voir la présence d'erreur, on appuie sur Ctrl+Shift+I ou F12
- 4) Si un caractère a un sens en JS, on ajoute un antislash devant « \ » pour qu'il soit pris comme un simple caractère.
- 5) Pour afficher du texte, on peut utiliser la fonction `alert(...)`

Exercice

Exercice 1

- a) Créer une page avec deux boutons qui auront le comportement suivant :
 - « Dialogue 1 » : Affiche le texte « Un oiseau ? Un avion ? »
 - « Dialogue 2 » : Affiche le texte « Non, c'est SUPERMAN ! »
- b) Si ce n'est pas déjà fait, faites-en sortent que vos boutons utilisent la même fonction « Dire(...) » avec des paramètres différents.

Exercice 2

La commande `prompt` affiche une fenêtre au l'utilisateur peut entrer une valeur (équivalent à `input` en python).

Pour concaténer deux textes, on utilise le « + ».

Exemple :

```
nombre = prompt("Choisis un nombre : ");  
alert("Tu as dit " + nombre)
```

Faites un programme qui demande votre nom (par exemple Dupont) puis qui vous répond « Bonjour Dupont, comment vas-tu ? »

Exercice 3

Faites un programme qui demande votre âge, puis qui vous dit si vous êtes mineur ou majeur.

TP - 2 : Changement de contenu

Un des grands intérêts du Javascript est de changer le contenu de la page sans avoir besoin de réactualiser la page en entier. Nous allons rester sur des exemples simples.

De plus, la fonction `prompt` n'est pas souvent utilisée. La preuve : quand on vous demande de vous connecter sur un site, vous saisissez votre identifiant et votre mot de passe dans des zones où vous pouvez saisir votre texte.

Nous allons voir d'autre type d'interaction que l'on peut faire en javascript.

1) Créer une page HTML « TP2.html » avec le code suivant :

```
<!DOCTYPE html>
<html>
<head>
  <title>TP - 2</title>
  <style>
    .visible{ display : bloc}
    .invisible {display : none}
  </style>
  <script type='text/javascript'>

    </script>
</head>
<body>
<input id="mon_texte" type="text" value="toto"/><br>
<div id="ma_zone"> Voici du contenu </div><br>
<div id="ma_zone2"> On va cacher ça !</div><br>

<button id="bnt1" >Copier</button><br>
<button id="bnt2" >Changer couleur</button><br>
<button id="bnt3" >Cacher</button>
<button id="bnt4" >Afficher</button><br>
</body>
</html>
```

Cette page contient :

- Du code CSS avec :
 - o Une classe `visible` qui rend visible un élément HTML.
 - o Une classe `invisible` qui cache un élément HTML.
- Un champ texte où l'utilisateur peut saisir du texte.
- Deux zones contenant du texte qui n'est pas modifiable.
- Une série de bouton.

Remarque : Chaque élément HTML possède un attribut `id` qui est unique sur la page. Il va nous permettre d'identifier l'élément en question dans le code HTML.

Copier du texte

Le premier bouton va copier le texte du champs texte dans la zone de texte non modifiable.

1) Ajouter dans la partie script en tête de page le code suivant :

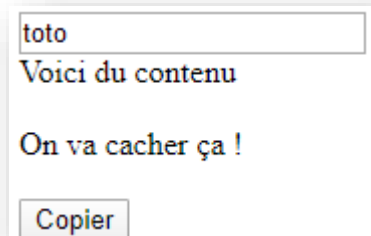
```
function copier(){
  champ_texte = document.getElementById("mon_texte");
  zone = document.getElementById("ma_zone");
  // Ici " champ_texte" est un composant HTML, une zone de texte.
  // Il faut récupérer son contenu
  texte = champ_texte.value;
  zone.innerText = texte;
}
```

Remarque : nous verrons dans la suite de l'activité plus précisément les éléments `input`. Gardez à l'esprit que

- on peut accéder au contenu d'une balise HTML avec le paramètre `innerText` (entre autre)
 - pour les `input`, on utilise le paramètre `value`
- 2) Ajoutez le code suivant dans les attributs du bouton `btn1` (le bouton « Copier »)

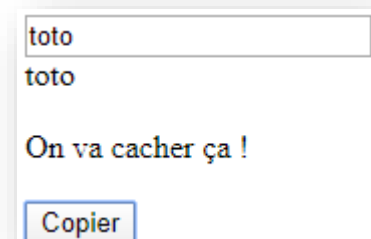
```
onclick="copier() "
```

- 3) Enregistrez et ouvrez votre page dans une navigateur

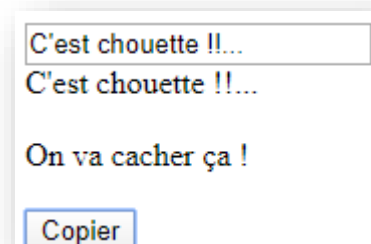


- 4) Cliquez sur le bouton « Copier »

Le texte a changé



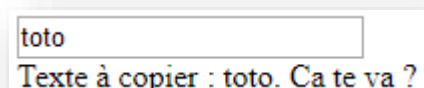
- 5) Modifiez le texte puis recommencez. Le texte a à nouveau changé.



Exercice - Aperçu

Faite un bouton qui au clic appellera une fonction `aperçu()` qui affiche le texte suivant :

« Texte à copier : mon texte. Ca te va ? »



Changement de couleur

Le deuxième bouton aura le comportement suivant :

- au survole de la souris, le champ texte deviendra rouge clair (code couleur `#FF8080`)
- au clic de la souris, le champs texte deviendra vert clair (code couleur `#80FF80`)

1) Dans la partie Javascript, ajoutez la fonction suivante :

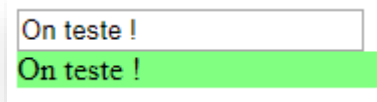
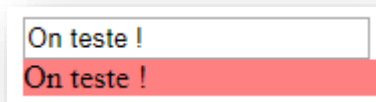
```
function changerCouleur(couleur){
    zone = document.getElementById("ma_zone");
    zone.style.backgroundColor=couleur;
}
```

2) Ajoutez les attributs `onclick` et `onmouseover` sur le bouton « btn2 » pour obtenir le code suivant (vous pouvez faire des retours à la ligne pour mieux voir les différents événements écoutés)

```
<button id="bnt2"
    onclick="changerCouleur('#80Ff80') "
    onmouseover="changerCouleur('#FF8080') ">
    Changer couleur
</button>
```

3) Enregistrez le code et actualisez la page HTML

4) Passez au dessus du bouton « Changer de couleur », puis cliquez dessus.



Exercice – Aperçu (suite)

Modifiez le code du bouton « Copier » pour avoir le comportement suivant :

- Quand la souris survole de bouton « Copier », on a un aperçu du texte à copier (en utilisant la fonction `aperçu` de l'exercice précédent)
- Quand un clic que le bouton « Copier », le texte est copié tout simplement (en utilisant la fonction `copier`).

Remarque : n'oubliez pas d'ajouter l'évènement `onmouseover` sur le bouton « Copier ».

Exercice – Masquer / Montrer

C'est une fonctionnalité souvent utilisée pour sur les sites web afficher ou masquer du contenu.

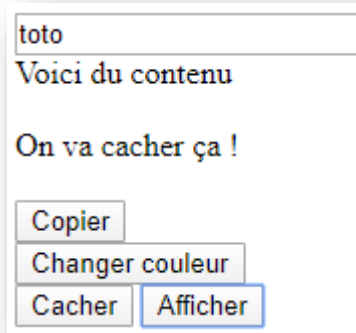
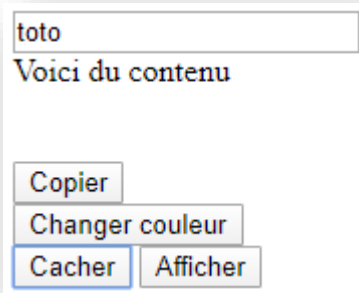
Pour cela on va changer dynamiquement les classes CSS appliquées aux éléments HTML

Ajoutez dans la partie script le code suivant :

```
function cacher(){
    zone_a_cacher = document.getElementById("ma_zone2");
    zone_a_cacher.className="invisible";
}

function montrer(){
    zone_a_monter = document.getElementById("ma_zone2");
    zone_a_monter.className="visible";
}
```

a) Modifiez le code des boutons « Cacher » et « Afficher » pour utiliser ces fonctions correctement



- b) Modifier le code des fonctions `cache` et `montrer` pour ne laisser que le bouton utile (le bouton « Afficher » si le texte est caché, et le bouton « Cacher » si le texte est affiché).

Indice : vous devez appliquer correctement les classes `visible` et `invisible` aux boutons.

Externaliser le contenu JS

Comme pour le CSS, nous pouvons faire appel à du code javascript directement dans un élément HTML. Comme pour le CSS nous pouvons faire un bloc de code en entête de la page HTML. C'est ce que qu'on a vu jusqu'à présent.

Comme pour le CSS, nous pouvons déporter le code Javascript dans un fichier à part.

- 1) Faites une copie de votre fichier « TP2.html » (par exemple « TP2 – Copier.html »)
- 2) Créez au même endroit que votre fichier « TP2.html » un répertoire « js »
- 3) Créez dans ce répertoire un fichier « tp2.js »
- 4) Déplacez tout le contenu du bloc script dans le fichier HTML (sans les balises `<script>`) dans le fichier « scriptTP2.js »
- 5) Remplacez les balises `<script>` par la ligne suivante :

```
<script src="js/tp2.js"></script>
```

- 6) Enregistrez le fichier HTML, et le fichier JS et actualisez votre page HTML.

Le site doit fonctionner comme avant. Pour tester, vous pouvez modifier le texte de la fonction `aperçu()`.

TP - 3 : Les formulaires

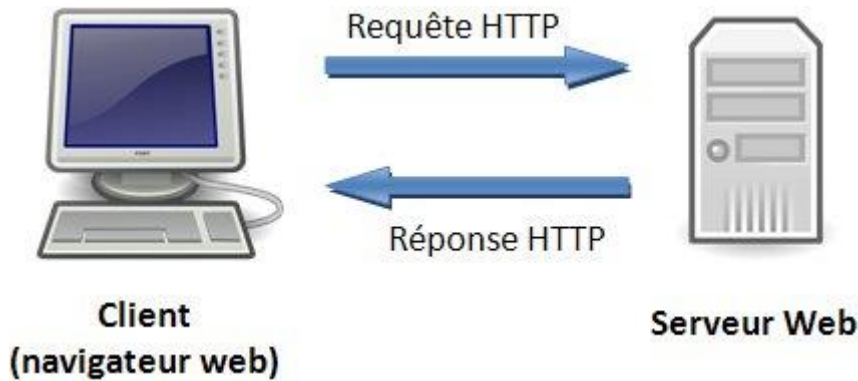
Introduction

Sur internet, on retrouve une configuration Client-Serveur.

- Le client demande à un serveur un service (le menu M13)
- Le serveur lui fournit (le menu M13 : 6 maki saumon-avocat avec 2 brochettes bœuf-fromage avec son riz)

L'information va toujours dans ce sens.

- 1) Le client fait une requête HTTP au profit d'un serveur.
- 2) Le serveur délivre la réponse HTTP.



En aucun cas le serveur initie la requête, pour des raisons de sécurité.

Le client peut utiliser deux méthodes pour transmettre sa requête et ses paramètres (menu B13 ou A24, ou chirashi S2...)

- Méthode **GET** : les informations sont lisibles clairement via l'URL (Le serveur marque sur un formulaire le menu choisi et l'envoi en cuisine)
- Méthode **POST** : Les informations sont cachées dans l'entête de la requête HTTP (le client donne son menu dans une enveloppe fermée à destination du cuisinier uniquement)

Nous allons voir comment mettre cela en place.

Méthode GET - Méthode POST

Pour commencer, nous allons utiliser un composant HTML prévue pour cela à savoir un formulaire (`<form>`). Celui-ci sera composé d'entrée pour la saisie (des `<input>`).

- 1) Ouvrez dans un éditeur de texte le fichier « TP3.html »

Nous allons regarder plus en détail deux éléments :

- `<form>` : le formulaire qui contient des zones de saisie (`<input>`) avec un paramètre `name` pour que l'utilisateur. Il possède deux paramètres importants :
 - o `method` : indique la méthode que l'on va utiliser, méthodes décrites plus haut. Nous allons voir les différences. Pour le moment la méthode qu'on utilisera sera la méthode GET
 - o `action` : action qui sera exécutée à la validation du formulaire. Dans notre cas, nous irons sur une page affichant les informations du formulaire récupérées par le serveur.
- `<input type="submit">` : bouton d'envoi du formulaire. Si vous appuyez sur « Entrée », l'action est exécutée.

Méthode GET

- 1) Ouvrez la page dans un navigateur.
- 2) Saisissez un identifiant et un mot de passe (pas les vôtres, ils apparaîtront en claires et peuvent être récupérés par le serveur)
- 3) Appuyez sur « Entrée »

Vous arrivez sur une page vous indiquant les différentes informations récupérées par les deux méthodes GET et POST.

- 4) Vérifiez les différentes informations récupérées. Ce sont les vôtres (plus une autre ^_^).

- 5) Regardez-en haut du navigateur l'URL de la page (l'endroit où vous entrez l'adresse de la page)
- 6) Modifiez les valeurs directement dans l'URL puis appuyez sur « Entrée »
- 7) Vérifiez que les informations ont bien changées

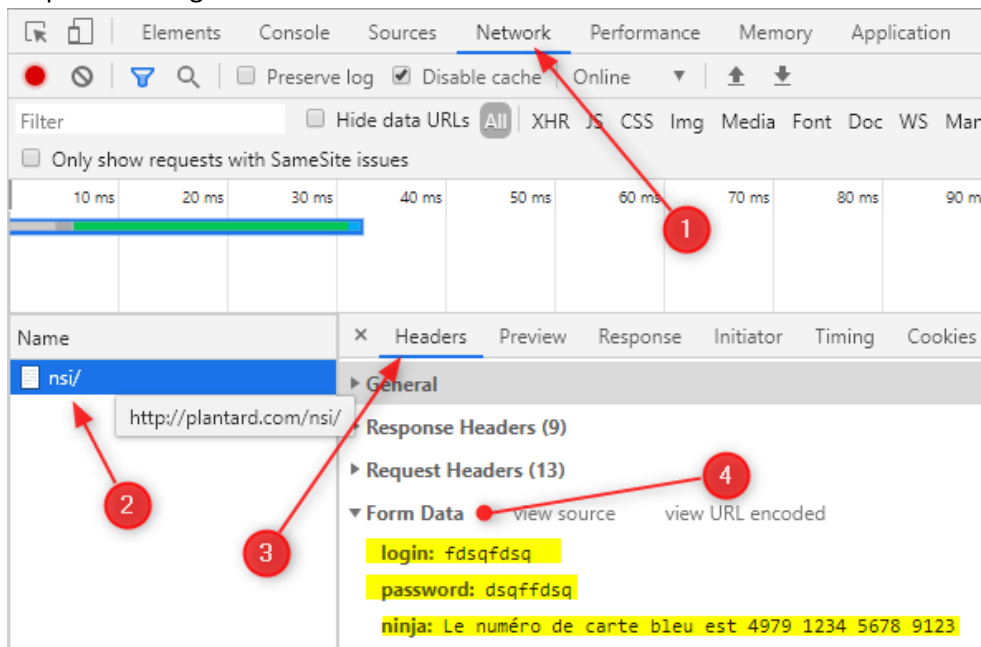
Si dans un autre onglet ou une autre fenêtre, vous recopiez l'URL de la page, vous aurez les mêmes informations d'afficher (celle si étant directement lié à l'URL)

Méthode POST

- 1) Dans le fichier « TP3.html », dans la balise `form`, modifiez la valeur de l'attribut `method` par `POST`.
- 2) Enregistrez et actualiser votre page.
- 3) Renseignez un identifiant et un mot de passe.
- 4) Appuyez sur « Entrée »
- 5) Vérifiez l'exactitude des données

Cette fois, aucune information n'est présente dans l'URL. Si vous ouvrez la page dans un autre onglet ou une nouvelle fenêtre. Les informations seront perdues.

- 6) Ouvrez la fenêtre des outils de développement
- 7) Cliquez sur l'onglet « Réseau » ou « Network »
- 8) Cliquez sur « nsi/ »
- 9) Cliquez sur l'onglet « En-tête » ou « Header »



Vous pouvez voir les informations passées en paramètre. Elles ne sont cependant pas modifiables.

- 10) Vérifiez l'exactitude des données.

Exercice : Compréhension du formulaire

- a) Repérez dans de code la partie qui envoie le numéro de carte bleu.
- b) Essayer d'entrer un mot de passe ayant moins de 6 caractères. Vous avez un message d'erreur et vous rester sur la page de saisie.
- c) Repérez dans le code la partie permettant de faire cette vérification
- d) Changer le code pour forcer la saisie d'un mot de passe d'au moins 8 caractères.
- e) En vous aidant d'internet, faite en sorte de ne valider que les identifiants ayant une longueur entre 5 et 10 caractères, et commençant par une majuscule.

Exercice : Reset

- a) Faites-en sorte que le bouton « Effacer » efface le contenu dans différents éléments saisis

Indication : Il faut changer en Javascript la valeur du paramètre `value`.